

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/KR04/002759

International filing date: 29 October 2004 (29.10.2004)

Document type: Certified copy of priority document

Document details: Country/Office: KR

Number: 10-2003-0097243

Filing date: 26 December 2003 (26.12.2003)

Date of receipt at the International Bureau: 26 November 2004 (26.11.2004)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

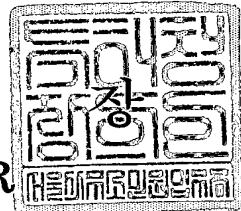
This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 : 10-2003-0097243
Application Number

출원년월일 : 2003년 12월 26일
Date of Application DEC 26, 2003

출원인 : 한국전자통신연구원
Applicant(s) Electronics and Telecommunications Research Inst

2004년 10월 12일



특허청

COMMISSIONER

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0056
【제출일자】	2003.12.26
【국제특허분류】	G06F
【발명의 명칭】	XML 처리 장치 및 이를 적용한 시스템의 XML 처리 방법
【발명의 영문명칭】	XML processing apparatus and XML processing method in the system adapting that
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2001-038378-6
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2001-038396-8
【발명자】	
【성명의 국문표기】	이강찬
【성명의 영문표기】	LEE,Kang Chan
【주민등록번호】	700519-1400819
【우편번호】	301-152
【주소】	대전광역시 중구 태평2동 버드내아파트 139-2501
【국적】	KR
【발명자】	
【성명의 국문표기】	전종홍
【성명의 영문표기】	JEON,Jong Hong
【주민등록번호】	690512-1025320

【우편번호】	305-761		
【주소】	대전광역시 유성구 전민동 엑스포아파트 209-1603		
【국적】	KR		
【발명자】			
【성명의 국문표기】	이원석		
【성명의 영문표기】	LEE, Won Suk		
【주민등록번호】	730609-1573311		
【우편번호】	302-162		
【주소】	대전광역시 서구 도마2동 경남아파트 114-506		
【국적】	KR		
【발명자】			
【성명의 국문표기】	이승윤		
【성명의 영문표기】	LEE, Seung Yun		
【주민등록번호】	670719-1025316		
【우편번호】	305-330		
【주소】	대전광역시 유성구 지족동 열매마을아파트 202-808		
【국적】	KR		
【발명자】			
【성명의 국문표기】	박기식		
【성명의 영문표기】	PARK, Ki Shik		
【주민등록번호】	580310-1552820		
【우편번호】	305-333		
【주소】	대전광역시 유성구 어은동 한빛아파트 101-602		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 이영필 (인) 대리인 이해영 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	6	면	6,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	9	항	397,000 원

【합계】	432,000 원
【감면사유】	정부출연연구기관
【감면후 수수료】	216,000 원
【기술이전】	
【기술양도】	희망
【실시권 허여】	희망
【기술지도】	희망
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】

【요약】

본 발명은 확장성 생성 언어(extensible mark-up language ; XML) 처리에 관한 것으로서, 더욱 상세하게는 독립적인 하드웨어를 기반으로 XML 처리 기능의 일부를 하드웨어적으로 처리하도록 하여 시스템의 부담을 줄이면서도 XML 처리 속도를 개선하는 XML 처리 장치 및 이를 적용한 시스템의 XML 처리 방법에 관한 것이다.

본 발명에 따른 XML 처리 장치는 XML 처리를 수행하기 위한 소프트웨어 실행시 필요한 변수, 값들을 저장하기 위한 제1메모리; XML 처리 기능의 일부를 하드웨어적으로 처리하는 하드웨어 처리 모듈; 상기 하드웨어 처리 모듈이 사용하는 제2메모리; 및 XML 처리기의 전반적인 제어를 담당하는 중앙처리장치를 포함하는 것을 특징으로 한다.

본 발명에 따른 XML 처리기는 독립적인 하드웨어를 기반으로 XML 처리 기능의 일부를 하드웨어적으로 처리하도록 함으로써 기존의 소프트웨어적으로 처리하는 방법에 비하여 XML처리 속도를 개선하고, 시스템의 부담을 줄이는 효과를 가진다.

【대표도】

도 4

【명세서】

【발명의 명칭】

XML 처리 장치 및 이를 적용한 시스템의 XML 처리 방법{XML processing apparatus and XML processing method in the system adapting that}

【도면의 간단한 설명】

도 1은 본 발명에 따른 XML 처리 장치와 종래의 소프트웨어에 의한 범용 XML 파서의 동작을 도식적으로 보이고 있다.

도 2는 본 발명에 따른 XML 처리 장치와 종래의 소프트웨어에 의한 특수 용도의 XML 파서의 동작을 도식적으로 보이고 있다.

도 3은 본 발명에 따른 XML 처리기의 처리 용량을 도식적으로 보이는 것이다.

도 4는 본 발명의 XML 처리기의 실시예를 보이는 것이다.

도 5는 도 4에 도시된 XML 처리 장치에 있어서 메모리 관리 기능을 하드웨어적으로 구현하는 방법을 도식적으로 보이는 것이다.

도 6은 도 5에 도시된 장치에서 메모리를 할당하는 절차를 보이는 흐름도이다.

도 7은 도 5에 도시된 장치에서 메모리를 재할당하는 절차를 보이는 흐름도이다.

도 8은 도 5에 도시된 장치에서 메모리를 반환하는 절차를 보이는 흐름도이다.

도 9는 본 발명에 따른 XML 처리기를 활용하는 시스템의 소프트웨어 계층도를 보이는 것이다.

도 10은 대량의 XML 처리를 위한 시스템의 소프트웨어 계층도를 보이는 것이다.

도 11은 본 발명에 따른 XML 처리 방법을 보이는 흐름도이다.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<12> 본 발명은 확장성 생성 언어(extensible mark-up language ; XML) 처리에 관한 것으로서, 더욱 상세하게는 독립적인 하드웨어를 기반으로 XML 처리 기능의 일부를 하드웨어적으로 처리하도록 하여 시스템의 부담을 줄이면서도 XML 처리 속도를 개선하는 XML 처리 장치 및 이를 적용한 시스템의 XML 처리 방법에 관한 것이다.

<13> XML은 인터넷상에서 구현되는 전자 상거래, 웹 포털들, 컨텐츠 서비스들 및 정보 처리 어플리케이션들을 위한 언어이다. XML 표준은 XML 문서(document)로 불리는 데이터 객체들의 클래스와 이러한 XML 문서들을 처리하는 방식에 대하여 기술하고 있다.

<14> XML 문서는 문자들로 구성되어 있으며, 그중 몇몇은 문자 데이터를 형성하며, 또 다른 몇몇은 마크업(mark-up)을 형성한다. 마크업은 시작 태그(tag)와 마침 태그의 쌍으로 구성되어 있으며, 문서 레이아웃의 설명 및 XML 문서들 사이의 논리적인 구조를 제공한다.

<15> 한편, XML 파서(parser)는 XML 문서의 조작을 용이하게 하기 위해 사용되는 소프트웨어 라이브러리로 파악할 수 있다. 대부분의 XML 파서들은 XML 문법과 호환하도록 구성된다.

<16> 종래의 범용적인 XML 파서들의 중요한 결점은 이들이 XML 표준들과 완벽하게 호환되도록 하기 위하여 비교적 큰 규모의 소프트웨어 구성 요소들을 요구한다는 것이다. 이에 따라 XML 문서들을 처리하는 시스템의 부담이 증가하는 문제점이 있다.

<17> 정보화 시대에서 인터넷의 중요성은 날로 증대하고 있으며 특히, 전형적인 PC 이외에 핸드폰, 디지털 가전, 텔레메틱스 단말기, PDA, 웹 TV 등의 다양한 임베디드 장치에서의 웹 사용

이 급격하게 증가하고 있다. 이를 임베디드 장치들은 전형적으로 PC에 비해 제한된 컴퓨팅 능력 및 메모리를 가지기 때문에 소프트웨어에 기반한 범용의 XML 파서들은 이를 임베디드 장치에 적합하지 않게 된다.

<18> 한편, 비PC 기반의 장치에 적합하도록 특정 용도에 맞춘 XML 파서들이 개발되고 있다. 그렇지만, 이러한 특정 용도에 맞춘 XML 파서들 조차도 기본적으로 소프트웨어 라이브러리로 구성되기 때문에 여전히 비PC 기반의 장치가 XML 처리를 수행하는 것에 부담을 가지게 된다.

<19> 이에 따라 보다 원활하게 XML 문서를 처리할 수 있는 방법이 요청되고 있다.

【발명이 이루고자 하는 기술적 과제】

<20> 본 발명은 상기의 요구에 부응하기 위하여 고안된 것으로서 독립적인 하드웨어를 기반으로 XML 처리 기능의 일부를 하드웨어적으로 처리하도록 하여 기존의 소프트웨어적으로 처리하는 방법에 비해 처리 속도를 개선하는 XML 처리 장치를 제공하는 것을 그 목적으로 한다.

<21> 본 발명의 다른 목적은 상기의 XML 처리 장치와 소프트웨어에 기반한 XML 처리기를 구비하는 장치에 있어서 효율적으로 XML 처리를 수행할 수 있는 방법을 제공하는 것에 있다.

【발명의 구성 및 작용】

<22> 상기의 목적을 달성하는 본 발명에 따른 XML 처리 장치는

<23> XML 처리를 수행하는 장치에 있어서,

<24> XML 처리를 수행하기 위한 소프트웨어 실행시 필요한 변수, 값들을 저장하기 위한 제1메모리;

<25> XML 처리 기능의 일부를 하드웨어적으로 처리하는 하드웨어 처리 모듈;

<26> 상기 하드웨어 처리 모듈이 사용하는 제2메모리;

<27> XML 처리기의 전반적인 제어를 담당하는 중앙처리장치를 포함하는 것을 특징으로 한다.

<28> 이하 첨부된 도면을 참조하여 본 발명의 구성 및 동작을 상세히 설명하기로 한다.

<29> 본 발명은 시스템의 중앙처리장치에 부담을 많이 주는 XML 처리를 보다 원활히 할 수 있도록 도와주는 XML 처리기를 제공한다.

<30> 기존의 XML 처리는 전적으로 소프트웨어로 구현되기 때문에 많은 양의 XML 처리를 수행할 때 CPU에 많은 부담을 주었다. 이는 시스템 전체의 성능을 저하시켜 시스템이 수행하는 다른 기능들에까지 영향을 주며, 결과적으로 시스템의 성능저하에 따른 경제적인 손실로 이어졌다.

<31> 또한, 임베디드 시스템 즉, 핸드폰, 디지털 가전, 텔레메틱스 단말기, PDA 등과 같이 CPU 성능이 PC에 비해 상대적으로 낮은 시스템에서의 XML 처리는 시스템의 큰 부담이다.

<32> 본 발명은 XML에 대한 처리를 독립적인 하드웨어를 기반으로 고속으로 처리하므로 위와 같은 문제를 해결할 수 있으며, 각 시스템의 특성에 맞게 적용될 수 있다.

<33> 도 1은 본 발명에 따른 XML 처리 장치와 종래의 소프트웨어에 의한 범용 XML 파서의 동작을 도식적으로 보이고 있다.

<34> 본 발명에 따른 XML 처리기(13)와 종래의 소프트웨어에 의한 범용 XML 파서(11)들은 도 1에 도시된 바와 같이 모두 같은 XML 문서(10)를 입력으로 받아 처리한 후 동일한 결과물(12)을 생성한다.

<35> 기존의 소프트웨어를 기반으로 한 범용 XML 파서(11)는 마이크로소프트의 MSXML(Microsoft XML Parser), IBM의 XML for C++, XML for java, 썬마이크로시스템의

JAXP(Java API for XML Processing) 등 W3C(World Wide Web Consortium)의 XML 버전 1.0을 완전히 지원하는 파서이다.

<36> XML 처리기(13)는 XML 처리의 일부분 특히, 메모리 관리에 관한 부분을 하드웨어적으로 처리하므로 기존의 소프트웨어를 기반으로 한 범용 XML 파서(11)에 비해 파싱 속도가 빠르다.

<37> 이에 따라 본 발명에 따른 XML 처리기(13)는 XML 처리시 CPU에 인가되는 로드를 줄여주어서, XML 문서를 자연스럽게 처리하면서도 다른 처리를 원활히 수행할 수 있도록 한다.

<38> 도 2는 본 발명에 따른 XML 처리 장치와 종래의 소프트웨어에 의한 특수 용도의 XML 파서의 동작을 도식적으로 보이고 있다.

<39> 본 발명에 따른 XML 처리기(17)와 종래의 소프트웨어에 의한 특수 용도의 XML 파서(15)들은 도 2에 도시된 바와 같이 모두 같은 XML 문서(14)를 입력으로 받아 처리한 후 동일한 결과물(16)을 생성한다.

<40> 도 3은 본 발명에 따른 XML 처리기의 처리 용량을 도식적으로 보이는 것이다. 도 3을 참조하면, XML 처리 기능이 필요한 각각의 시스템들이 요구되는 XML 처리 용량에 따라서 본 발명에 따른 XML 처리기의 처리 용량을 다양하게 적용될 수 있다는 것을 알 수 있다.

<41> 도 3에 있어서, 화살표(34) 방향은 XML 처리 용량이 많아지는 방향을 의미한다. 도 3을 참조하면, 본 발명에 따른 XML 처리기는 처리 용량, 적용될 시스템 특성에 따라서 적합한 형상을 가질 수 있다는 것을 알 수 있다.

<42> 도 4는 본 발명의 XML 처리기의 실시예를 보이는 것이다. 도 4에 도시된 장치는 XML 처리기의 전반적인 제어를 담당하는 중앙처리장치(40), 중앙처리장치(40)가 특정 기능을 수행하

기 위해 소프트웨어의 실행 및 소프트웨어 실행 시 필요한 변수, 값들을 저장하기 위한 메모리(41), XML 처리 기능의 일부를 하드웨어적으로 처리하는 하드웨어 처리 모듈(42), 그리고 하드웨어 처리 모듈(42)이 사용하는 메모리(43)가 있다. 또한 이 모든 요소들이 데이터를 주고 받기 위한 버스(44)를 통해서 연결되어 있다.

<43> 도 4는 XML 처리 기능들 중 특정 부분이 하드웨어적으로 구현될 수 있음을 보여준다. 예를 들면, 소프트웨어 파서들의 성능에 가장 영향을 많이 미치는 것이 파싱에서 사용되는 메모리 할당, 반환, 재할당에 대한 처리 부분이다.

<44> 도 4를 참조하면, 본 발명에 따른 XML 처리기는 XML 처리 성능을 개선하기 위하여 메모리 관리 기능 부분을 하드웨어적으로 구현할 수 있다. 또한, 본 발명에 따른 XML 처리기는 메모리 관리 기능 이외에도 XML 처리에 있어서 많이 사용되는 XML DTD(Data Type Definition) 혹은 XML 스키마에 대한 상태 머신(State Machine)을 하드웨어적으로 구현할 수도 있다.

<45> 도 5는 도 4에 도시된 XML 처리 장치에 있어서 메모리 관리 기능을 하드웨어적으로 구현하는 방법을 도식적으로 보이는 것이다.

<46> XML 처리시 XML의 엘리먼트들은 메모리상의 하나의 노드로 표현되며, 이를 노드는 서로 관계(relation)를 갖고 트리 형태를 띠게 된다. 따라서 이를 노드를 위한 메모리 할당, 재할당, 반환에 대한 처리가 필요하다.

<47> 노드에 대한 메모리 관리를 위해 가장 중요하게 고려해야 할 것은 노드에서 유지해야 하는 정보들의 구성과 데이터의 크기가 유동적인 정보에 대한 처리이다.

<48> 노드가 유지해야 하는 정보는 XML 문서 처리를 위한 W3C 표준 중의 하나인 DOM(Document Object Model)에서 정의하고 것을 따르는 것이 좋다. DOM은 표준이

므로 대부분의 XML 파서들이 이를 지원하고 있어, 다양한 형태의 기존 파서와의 연동을 보다 쉽게 할 수 있기 때문이다.

<49> DOM에서 노드가 유지해야 할 정보로는 노드 이름, 노드 값, 노드 타입, 부모노드, 자식 노드들, 첫번째 자식노드, 마지막 자식 노드, 전 시블링, 후 시블링 애트리뷰트(attribute) 값 등이다. 보다 자세한 사항은 W3C의 DOM 표준을 참고하면 된다.

<50> 이 정보들 중에서 데이터의 크기가 유동적인 정보는 노드 타입이 PCDATA인 경우의 노드 값이다. 이외에 데이터 값을 갖는 노드 이름, 타입, 애트리뷰트 값 등은 크기가 크게 유동적이지 않으므로 적정한 사이즈를 고정으로 해서 처리해도 크게 문제가 없다. 또한 나머지 정보들은 포인터 정보이므로 주소정보를 저장할 수 있으면 된다.

<51> 이러한 특징들을 만족시키기 위해 하드웨어를 이용한 메모리 관리 기능 부분은 도 5에 도시된 바와 같이 Node Usage Check Table(301), Node Table(302) 그리고 실제 데이터를 유지하기 위한 Memory(303)로 구성될 수 있다. Node Usage Check Table(301)은 노드에서 유지해야 할 데이터의 크기에 따라서 여러 개의 큰 블록으로 나누어져 있으며, 큰 블록 안에 있는 각 칸은 대응되는 Node Table(302)의 사용여부를 표시한다. Node Table(302)은 각 노드가 유지해야 하는 모든 정보들 즉, 노드 이름, 노드 타입, 부모노드, 자식노드들 등을 관리하고 있는 테이블이다. 그러나 이 테이블은 실제 값을 유지하는 것이 아니고 값을 가지고 있는 메모리의 주소나 Node Table(302)의 주소를 유지한다. 따라서 테이블의 각각의 칸이 고정된 크기로 되어 있어 있을 수 있다.

<52> Memory(303)는 RAM(Random Access Memory)이 될 수 있으며 실제적인 값을 유지하기 위해 사용된다. Node Usage Check Table(301)과 Node Table(302)은 ASIC(Application-Specific Integrated Circuit) 형태로 구현 가능하다.

<53> 참고로, 도 5에 도시된 Node Usage Check Table(301)와 Node Table(302)은 서로의 관계가 고정되어 있다. 즉, Node Usage Check Table(301)에서 사용할 메모리 위치가 결정되면 Node Table에서 사용할 블록 위치가 자동으로 결정되는 것이다. 그리고 Node Table(302)은 실제 값을 갖는 부분이 아니고 실제 정보를 갖고 있는 Memory(303) 상의 주소를 저장하며, 이 또한 성능을 최대한 빠르게 하기 위해 고정된 위치로 맵핑 될 수 있도록 하는 것이 좋다.

<54> 그러나 실제로 Node Table(302) 상에서 실제 값을 갖는 정보는 소수이며, 가변성이 큰 정보는 거의 없으므로 메모리의 크기를 적정하게 설정하면 메모리의 낭비가 크지 않게 될 것이다. 또한, Node Table(302)에 저장될 주소는 Node Table 상의 다른 블록(즉, 다른 Node)에 대한 정보일 수도 있다.

<55> 도 6은 도 5에 도시된 장치에서 메모리를 할당하는 절차를 보이는 흐름도이다.

<56> 먼저 할당할 메모리의 크기 정보를 포함한 노드 정보를 얻는다.(310과정)

<57> XML 파싱을 하면서 DOM 트리를 만들기 위해 필요한 메모리 할당을 위해서는 할당할 메모리의 크기 정보를 포함한 노드 정보가 필요하다. 이 노드 정보를 기반으로 적절한 데이터 크기를 수용할 수 있는 메모리가 있는지 Node Usage Check Table에서 이용 가능 검사(311)을 해서 (311과정), 있으면 Node Usage Check Table(301)에 사용 표시를 하고(313과정), 없다면 오류처리를 한다.(312과정)

<58> Node Usage Check Table(301)과 Node Table(302) 사이의 관계가 고정되어 있으므로, Node Usage Check Table(301)에서 사용할 메모리의 위치를 결정하면 Node Table 상에서 사용할 부분의 주소를 자동으로 알 수 있다. 따라서 해당 Node Table의 주소 반환(314)를 수행해서 메모리 할당 수행은 끝이 난다.(314과정)

<59> 도 7은 도 5에 도시된 장치에서 메모리를 재할당하는 절차를 보이는 흐름도이다.

<60> 메모리의 재할당은 좀더 큰 데이터나 좀더 적은 데이터를 처리하기 위해 적절한 크기를 갖는 메모리를 할당하는 것이다.

<61> 먼저 재할당 정보를 얻는다.(320과정)

<62> 재할당 정보는 현재 사용하고 있는 노드 정보와 원하는 메모리 크기를 포함하며, 이를 기반으로 Node Usage Check Table(301)에서 재할당에 필요한 영역 검사를 수행해서 가능 여부를 결정한다.(321과정)

<63> 만일 메모리 재할당이 가능하다면 새로 사용한 부분에 대해 Node Usage Check Table(302)에 사용 표시를 하고(323과정), 할당 받은 Node Table(302)에 원본 Node의 정보를 복사한다.(324과정) 그 후 기존에 사용하던 것은 필요하지 않으므로 Node Usage Check Table(301)에서 원본 노드에 해당하는 칸 사용안함 표시를 하고, (325과정) 마지막으로 재할당한 Node Table(302)의 주소 반환(326)을 수행한다.

<64> 도 8은 도 5에 도시된 장치에서 메모리를 반환하는 절차를 보이는 흐름도이다.

<65> 먼저, 사용 노드 정보를 얻는다.(320과정)

<66> 사용 노드 정보(330)을 이용하여 해당 Node Usage Check Table(301)을 찾는다.(331과정)

<67> 찾은 Node Usage Check Table(301)에서 노드에 해당하는 칸에 사용안함 표시를 한다.(332과정)

<68> 도 9는 본 발명에 따른 XML 처리기를 활용하는 시스템의 소프트웨어 계층도를 보이는 것이다. 본 발명에 따른 XML 처리기를 활용하는 시스템(51)과 XML 처리기(30) 간의 인터페

이스 방식(56)은 다양한 방법이 사용 가능하다. 예를 들면, PCI, USB 등과 같은 표준 방식들이 사용될 수 있으며, 이때 주의해야 할 것은 인터페이스의 속도와 안정성이다.

<69> 시스템 측의 디바이스 드라이버(55) 계층은 XML 처리기(30)와 데이터를 주고받기 위한 가장 하위 레벨의 프로그램이다. XML 파서 API(Application Programming Interface, 54)가 디바이스 드라이버(55)의 상위에 있으며, 응용 프로그램(52)은 XML 파서 API(54)를 이용하여 XML 처리를 수행한다. 여기서, 언어 적용 계층(53)은 응용 프로그램 개발시 언어의 독립성을 유지할 수 있도록 하기 위한 것이다. 언어 적용 계층(53)은 언어별로 각각 지원되어야 한다.

<70> 도 10은 대량의 XML 처리를 위한 시스템의 소프트웨어 계층도를 보이는 것이다. 도 10에 도시된 시스템(60)의 전반적인 구조는 도 5에 도시된 시스템(50)과 거의 같으며, 단지 소프트웨어 XML 파서(62)가 포함되었다.

<71> 이는 보다 효과적으로 XML 정보를 처리하기 위해, 처리 조건에 따라서 소프트웨어적으로 처리하는 것이 유리한 경우는 소프트웨어 XML 파서(62)가 처리하고 그렇지 않은 경우는 XML 처리기(30)가 처리하는 구조이다.

<72> 이에 대한 구분은 XML 파서 API 계층(61)에서 결정한다. 이에 대한 처리 조건 및 절차가 도 11도를 참조하여 설명된다.

<73> 도 11은 본 발명에 따른 XML 처리 방법을 보이는 흐름도이다. 도 11은 도 4 내지 도 6을 참조하여 설명된 본 발명에 따른 XML 처리기 및 소프트웨어 기반의 XML 처리기를 가지는 시스템에 있어서의 XML 처리 방법을 보인다.

<74> 먼저, 입력된 XML 파일의 크기를 검사한다.(65과정)

<75> XML 파일 크기 검사(65)를 하는 이유는 특정 크기 이상의 XML 파일을 처리 하는 것은 제한된 메모리를 가지고 있는 XML 처리기의 전반적인 성능에 영향을 줄 수 있기 때문이다. 따라서 이런 경우는 소프트웨어 XML 파서를 이용하여 처리한다. (64과정)

<76> 입력된 파일의 크기가 설정된 크기보다 작다면, 다음은 XML 처리 결과로 파싱 트리 필요 여부를 검사한다.(66과정) 파싱 트리 필요 여부를 검사하는 이유는 XML 처리 후 트리를 결과로 제공할 필요가 없으면 소프트웨어 XML 파서로 처리한 성능과 XML 처리기로 처리한 성능이 크게 차이가 없으므로 소프트웨어로 처리하는 것이 바람직하다는 것 때문이다.

<77> XML 처리 결과로 트리가 필요 없을 경우에는, 소프트웨어 XML 파서로 처리한다.(64과정)
)

<78> XML 처리 결과로 트리가 필요한 경우에는, 빠른 처리 필요 검사(67)를 한다. 빠른 처리 필요 검사를 하는 이유는 빠른 처리가 필요하지 않은 파일은 소프트웨어적으로 처리하고, 빠른 처리가 필요한 파일을 XML 처리기를 이용하여 처리하는 것이 좋기 때문이다.

<79> 빠른 처리가 필요하지 않은 경우에는 소프트웨어 처리기에 의해 처리하고(64과정), 빠른 처리가 필요한 경우에는 XML 처리기를 이용하여 처리한다.(68과정)

<80> 본 발명에 따른 XML 처리기는 XML 처리를 필수적으로 요구하는 다양한 시스템에 적용 가능하며, 독립적인 하드웨어를 기반으로 XML 처리 기능의 일부를 하드웨어적으로 처리하도록 함으로써 기존의 소프트웨어적으로 처리하는 방법에 비하여 처리 속도를 개선한다.

<81> 본 발명에 따른 XML 처리기가 적용된 시스템에서는 시스템의 중앙처리장치가 XML에 대한 처리 로드를 부담하지 않아도 되므로 시스템의 성능이 크게 향상 되며, 컴퓨팅 파워가 약한 소형의 임베디드 디바이스들의 경우에는 보다 자연스럽게 XML을 처리할 수 있게 된다.

<82> 본 발명에 따른 XML 처리기는 XML 처리 성능을 크게 요구하는 시스템에서부터 XML에 대한 처리 성능 요구는 크지 않지만 XML에 대한 처리가 부담이 되는 시스템에서 활용될 수 있다.

<83> 즉, 다량의 XML 처리를 요구하는 전자상거래 서버에서부터 디지털 가전, 텔레메티кс 단말, 이동전화 단말 등 소형의 임베디드 시스템까지 본 발명을 포함할 수 있다.

<84> 컴퓨팅 파워가 약한 소형의 임베디드 시스템에서 사용되는 파서는 부담스러운 XML 처리를 줄이기 위해 1개 이상의 지정된 XML DTD(Document Type Definition)이나 스키마를 지원하거나 XML 표준의 일부분만을 지원할 수 있도록 하는 것이 일반적인 방식이다. 그러나 이와 같은 경우에도 보다 자연스러운 XML 처리를 위해 본 발명에 따른 XML 처리기를 포함하도록 할 수 있다.

<85> 본 발명에 따른 XML 처리기의 형태는 서버를 위해서는 PCI 카드 형태, 일반 테블릿 PC를 위해서는 USB 스틱 형태, 그리고 스마트폰이나 디지털 가전 등을 위해서는 SoC(System on Chip) 형태가 될 수도 있다. 이는 XML 처리 용량과 시스템의 특성에 대한 고려를 통하여 적합한 형태로 적용될 수 있음을 의미하며, 또한 시스템과의 인터페이스도 각 시스템의 특성에 맞도록 적용될 수 있음을 의미한다.

【발명의 효과】

<86> 상술한 바와 같이 본 발명에 따른 XML 처리기는 독립적인 하드웨어를 기반으로 XML 처리기능의 일부를 하드웨어적으로 처리하도록 함으로써 기존의 소프트웨어적으로 처리하는 방법에 비하여 XML처리 속도를 개선하고, 시스템의 부담을 줄이는 효과를 가진다.

【특허청구범위】**【청구항 1】**

XML 처리를 수행하는 장치에 있어서,

XML 처리를 수행하기 위한 소프트웨어 실행시 필요한 변수, 값들을 저장하기 위한 제1메모리;

XML 처리 기능의 일부를 하드웨어적으로 처리하는 하드웨어 처리 모듈;

상기 하드웨어 처리 모듈이 사용하는 제2메모리;

XML 처리기의 전반적인 제어를 담당하는 중앙처리장치를 포함하는 XML 처리 장치.

【청구항 2】

제1항에 있어서, 상기 하드웨어 처리 모듈은 XML 처리 기능들 중 XML 파싱에서 사용되는 메모리 할당, 반환, 재할당에 대한 처리를 수행하는 모듈임을 특징으로 하는 XML 처리 장치.

【청구항 3】

제2항에 있어서, 상기 하드웨어 처리 모듈은

노드 및 이들 노드들간의 트리 관계로 표현되는 XML의 엘리먼트들에 대한 메모리 할당, 재할당, 반환에 대한 처리를 수행하며,

여러 개의 블록으로 나뉘어져 있으며 각각의 블록이 대응되는 Node Table의 사용여부를 표시하는 Node Usage Check Table;

각 노드가 유지해야 하는 모든 정보들 즉, 노드 이름, 노드 타입, 부모노드, 자식노드 들 등을 관리하는 Node Table; 및

상기 Node Table의 각 요소들의 실제 내용을 저장하는 Node Memory를 포함하는 것을 특징으로 하는 XML 처리 장치.

【청구항 4】

제3항에 있어서, 상기 Node Table은 상기 Node Memory의 어드레스를 가지는 것을 특징으로 하는 XML 처리 장치.

【청구항 5】

제1항에 있어서, 상기 하드웨어 처리 모듈은 XML DTD(Data Type Definition)에 대한 처리를 수행하는 모듈임을 특징으로 하는 XML 처리 장치.

【청구항 6】

제1항에 있어서, 상기 하드웨어 처리 모듈은 XML 스키마에 대한 상태 머신(State Machine)을 수행하는 모듈임을 특징으로 하는 XML 처리 장치.

【청구항 7】

독립적인 하드웨어를 기반으로 XML 처리를 수행하기 위한 제1 XML 처리기 및 소프트웨어를 기반으로 XML 처리를 수행하기 위한 제2XML 처리기를 가지는 시스템의 XML 처리 방법에 있어서,

처리하고자 하는 XML 파일의 크기를 검사하는 과정; 및

처리하고자 하는 XML 파일의 크기가 설정된 크기보다 큰 경우에는 상기 제2XML 처리기에 의해 XML 처리를 수행하는 과정; 및

처리하고자 하는 XML 파일의 크기가 설정된 크기보다 크지 않은 경우에는 상기 제1XML에 의해 XML 처리를 수행하는 과정을 포함하는 XML 처리 방법.

【청구항 8】

제7항에 있어서,

XML 처리를 수행한 후 트리가 필요한 지의 여부를 검사하는 과정;

트리가 필요하지 않을 경우에는 상기 제1XML에 의해 XML 처리를 수행하는 과정; 및

트리가 필요할 경우에는 상기 제1XML 처리기에 의해 XML 처리를 수행하는 과정을 더 구비하는 것을 특징으로 하는 XML 처리 방법.

【청구항 9】

제7항에 있어서,

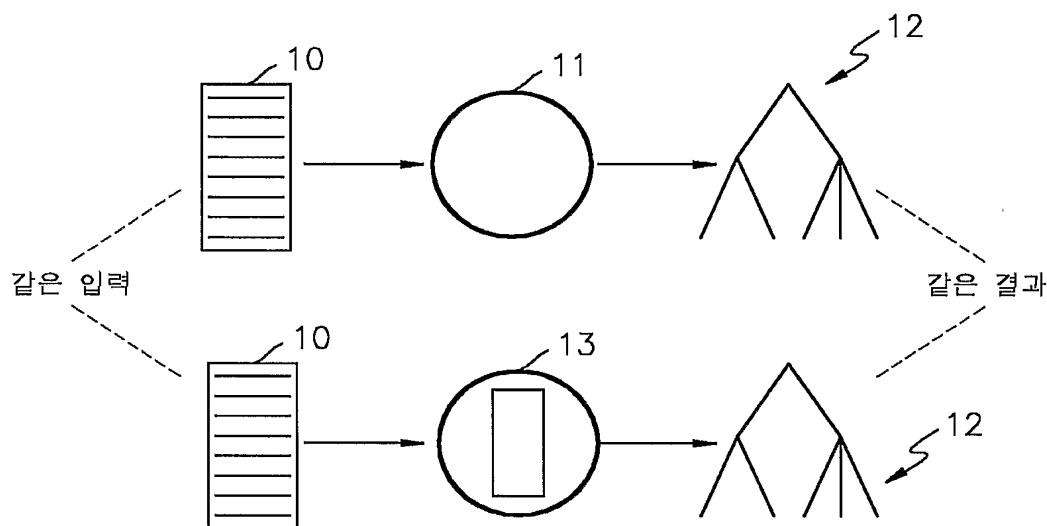
빠른 XML 처리가 필요한 지의 여부를 검사하는 과정;

빠른 XML 처리가 필요하지 않을 경우에는 상기 제2XML에 의해 XML 처리를 수행하는 과정; 및

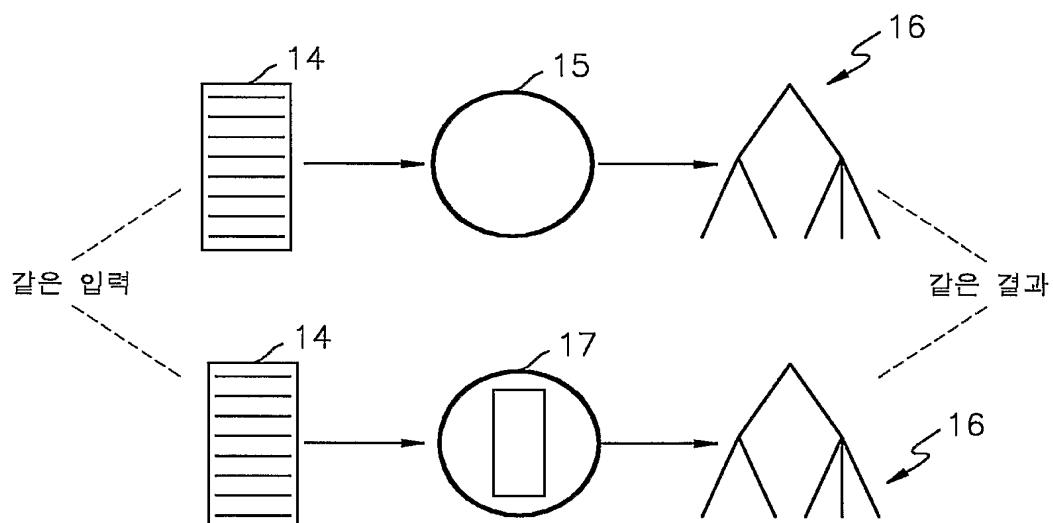
빠른 XML 처리가 필요할 경우에는 상기 제1XML 처리기에 의해 XML 처리를 수행하는 과정을 더 구비하는 것을 특징으로 하는 XML 처리 방법.

【도면】

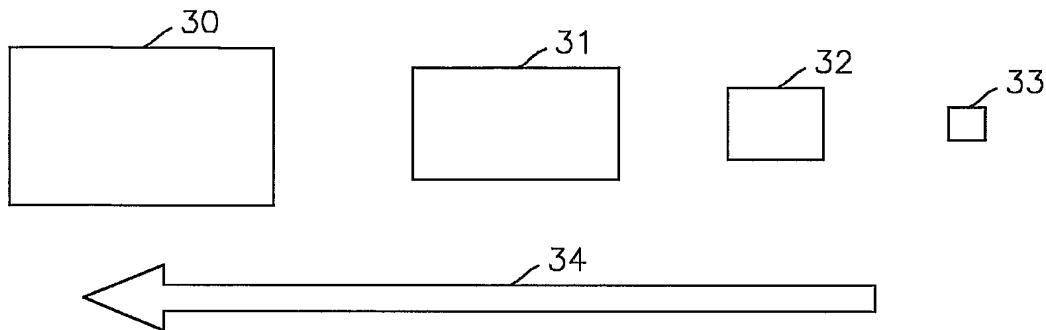
【도 1】



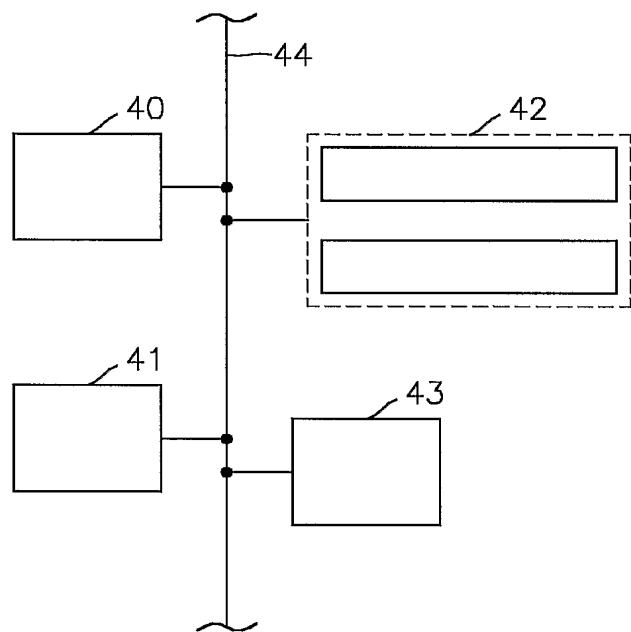
【도 2】



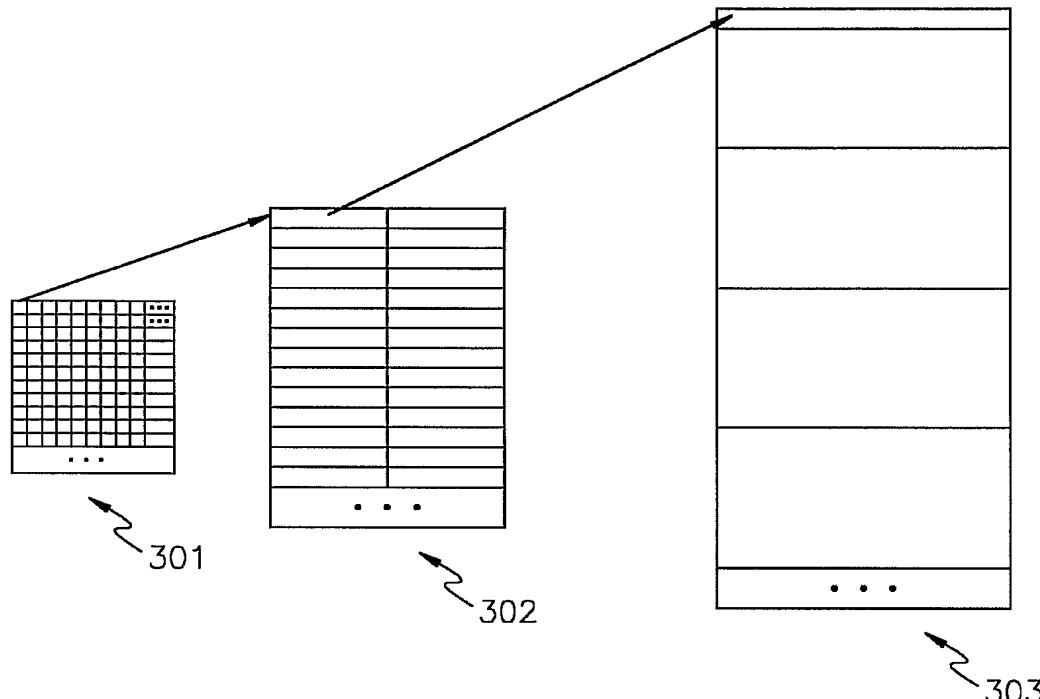
【도 3】



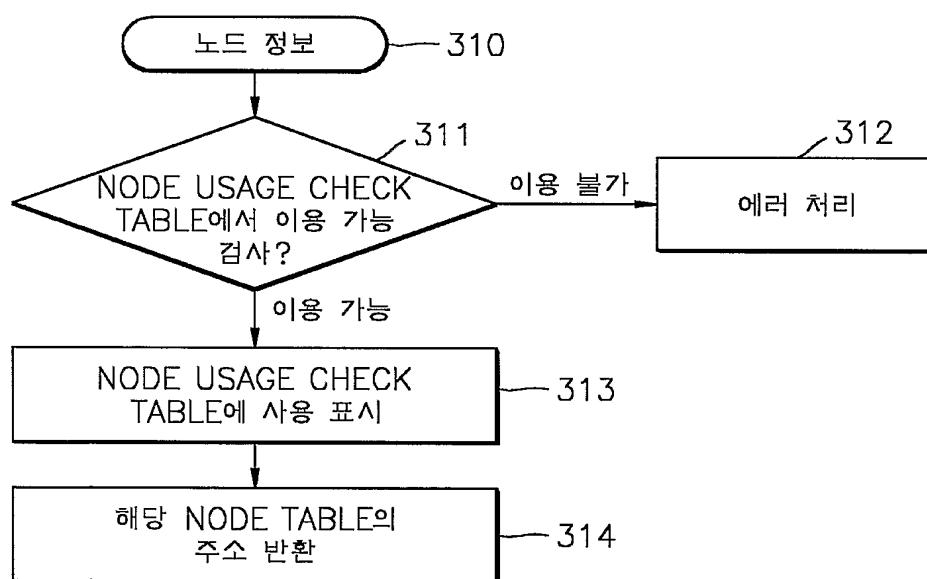
【도 4】



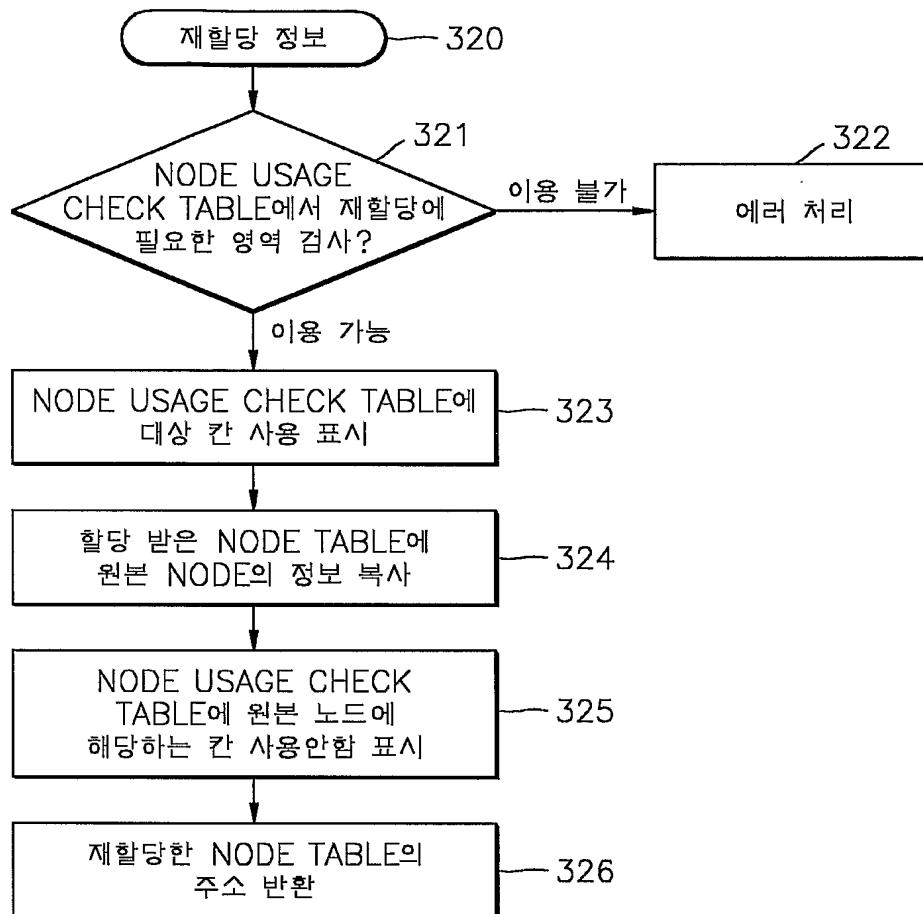
【도 5】



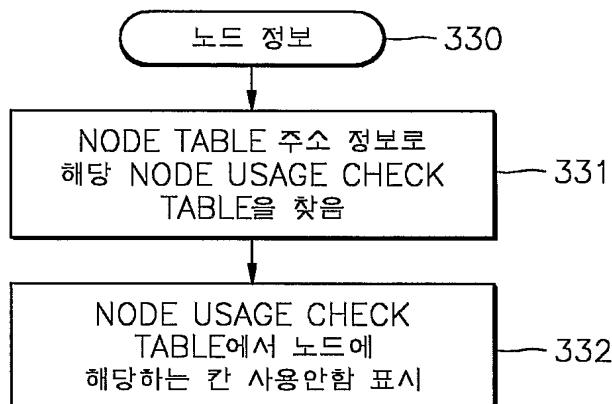
【도 6】



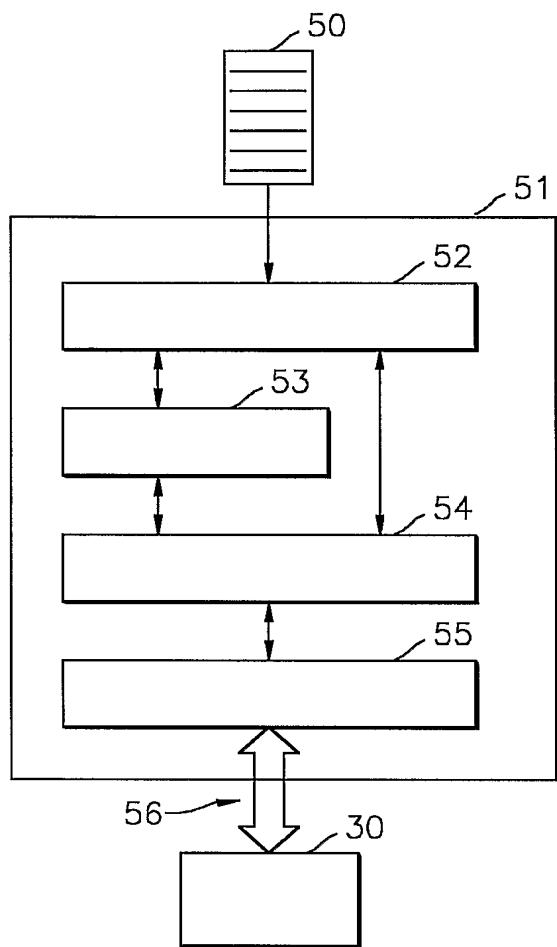
【도 7】



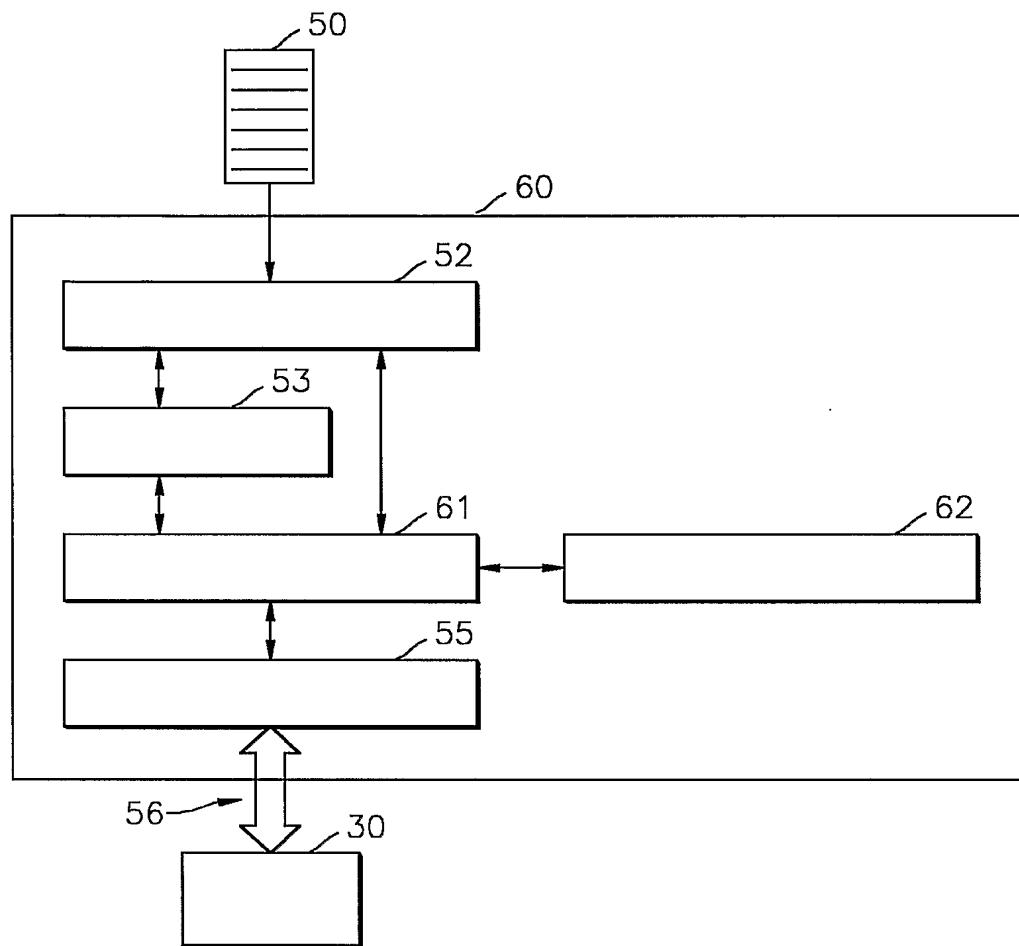
【도 8】



【도 9】



【도 10】



【도 11】

